DETAILED ACTION

1. This action is in response to Telephone Interview dated 02/18/2010 and Amendment/Req. Reconsideration-After Non-Final Reject filed 02/24/2010.

No Claims are amended.

Claims 1-20 are presented for examination have been examined on merits and are pending in this application.

Response to Amendment/Argument

2. Applicant's arguments filed on 02/24/2010 have been carefully considered, resulted in better understanding the claims and are persuasive hence previous rejection withdrawn.

However based on updated search new ground(s) of rejections are presented in this Non-Final Action.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

3. Independent Claims 1, 11 and -20 and its dependent claims rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement.

The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Claims recite "an inter-node communication network". There is no adequate support or description in the disclosure.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. **Claims 1-20** rejected under 35 U.S.C. 103(a) as being obvious over U.S. Patent No. 7334102 to Conway; Patrick, (hereinafter "Conway") and in view of U.S. Patent No. 6542513 to Franke; Hubertus et al., (hereinafter "Franke") and U.S. Patent Publication No. 20040260832 to Kota, Rajesh et al., (hereinafter "Kota").

As regards to Claim 1, *A method for communicating data from a first compute node of a computer system to a second node of the computer system, the computer system comprising multiple compute nodes, including the first and second compute nodes, interconnected by an inter-node communication network, the method comprising*

(Conway as stated in col. 2, lines 25-33, lines 44-45, col. 3, lines 14-25, discloses, A method is also provided for use in a non-uniform memory access (NUMA) data processing system for guaranteeing fair access to a lock variable among a plurality of processor nodes. A first access request with a lock acquire attribute for the lock variable is received from a first processor node. A second access request with a lock acquire attribute for the lock variable is received from a second processor node within a predetermined length of time after receiving the first access request. FIG. 1 illustrates a block diagram of a non-uniform memory access (NUMA) system 10 having eight nodes 100-107. The connections are effected by means of a coherent form of a communication protocol known as "HyperTransport". HyperTransport is one possible inter-node communication protocol):

placing the data on a full-duplex packetized interconnect directly connecting a CPU of the first compute node to a first network interface of the first compute node the first network interface directly connected to the inter-node communication network; receiving the data at the first network interface (Conway as stated in col. 4, lines 30-32, lines 44-49, col. 5, lines 4-16, discloses, FIG. 3 illustrates a block diagram of processing node 102 having balanced spinlock support for use in NUMA system 10 of FIG. 1. Microprocessor 300 includes generally a central processing unit (CPU) 302, a memory controller 304, a crossbar switch labeled "XBAR" 306, and three communication link controllers 308, 310, and 312 each labeled "HT" for HyperTransport. XBAR 306 is a switching/multiplexing circuit designed to couple together the buses internal to microprocessor 300. HT link controllers 308, 310, and 312 are coupled to devices

external to microprocessor 300 over corresponding input and output channels. Each of HT link controllers 308, 310, and 312 complies with the physical interface specified in the HyperTransport I/O Link Specification. In node 102 HT link controllers 308 and 310 function as coherent links that communicate with nodes 101 and 104 of FIG. 1, respectively, using a coherent form of HyperTransport. In response to decoding and executing an atomic instruction, CPU 302 presents the RdBlkM packet with the lock acquire attribute to one of HT link controllers 308 and 310 that will be part of the path back to the home node of the data element);

and, transmitting the data from the first network interface to a second network interface of the second compute node by way of the inter-node communication network (Conway as stated in col. 5, lines 23-29, lines 44-49, lines 63-67, discloses, Other nodes in system 10 assume the same basic configuration as node 102. However, nodes 1, 3, 4 and 6 use all three available HT link controllers for coherent HyperTransport links between adjacent processor nodes. Remaining nodes 0, 2, 5, and 7 have an additional HT link controller available for use as a host bridge for connection to I/O devices using non-coherent HyperTransport links. Cmd[5:0] is the command field which defines the packet type, and is equal to 000110 for the RdBlkM packet. SrcUnit[1:0] identifies the unit within the source node which generated the request. DstUnit[1:0] identifies the unit within the destination node to which this packet should be routed).

Conway as described in Fig. 3, HT links are separate input output packet links and HT link controller connecting directly to CPU through XBAR. I/O device 340 is an input/output device that, for example, implements the local area network communication

protocol standardized by the Institute of Electrical and Electronics Engineers (IEEE) under the auspices of the IEEE 802.3 committee, commonly referred to as "Ethernet". However other types of I/O functions are possible as well. Further as stated in publication “Hpertransport Technology Consortium; HyperTransport I/O Link Specification, Revision 1.03; Oct. 10, 2001; pp. 17-21”, HyperTransport technology is a packet-based link implemented on two independent unidirectional sets of wires. The HyperTransport link is nominally point-to-point and connects two devices. Chains of HyperTransport links can also be used as an I/O channel, connecting I/O devices and bridges to a host system.

Hence it is obvious that the HT links are Full-Duplex packetized links and connect first CPU directly to HT link controller available for use as a host bridge for connection to I/O devices using non-coherent HyperTransport links, which examiner considers being Network Interfaces.

As regards to Claim 2, *A method according to claim 1 wherein the first network interface and the CPU are the only devices configured to place data on the packetized interconnect* (Conway as stated in col. 5, lines 56-67, col. 6, lines 1-11, discloses, FIG. 4, which illustrates an encoding table 400 of a HyperTransport packet that can be used to form an access request packet with a lock acquire attribute. The various fields associated with packet 400 will now be described. Cmd[5:0] is the command field which defines the packet type, and is equal to 000110 for the RdBlkM packet. SrcUnit[1:0] identifies the unit within the source node which generated the request. DstUnit[1:0]

identifies the unit within the destination node to which this packet should be routed. Examples of units in this context are processors, memory controllers and bridges to non-coherent links. DstNode[2:0] identifies the node to which this packet should be routed. Depending on the packet, this field may identify either the source or target node for the transaction associated with this packet. SrcNode[2:0] identifies the original source node for the transaction. SrcTag[4:0] is a transaction tag which together with SrcNode and SrcUnit uniquely identifies the transaction associated with this packet. Each node can have up to 128 transactions in progress in the system. Addr[39:3] represents the 64-byte block address accessed by the request).

Thus it is apparent that CPU and HT link controller available for use as a host bridge for connection to I/O devices using non-coherent HyperTransport links are the only devices which place RdBlkM packet on the HT links.

As regards to Claim 3, *A method according to claim 1 comprising transmitting the data from the first network interface to the second computer node by way of a full-duplex communication link of the inter-node communication network* (Examiner uses same rational of Claim 1 to reject Claim 3 as internode communication is through HT links which is full-duplex as explained in claim 1).

As regards to Claim 4, *A method according to claim 3 comprising passing the data through a buffer at the first network interface before transmitting the data* (Conway as stated in col. 8, lines 28-31, discloses, In addition to using the new LM, LO and L

directory states, memory controller 304 may implement other techniques for determining when the lock variable has been released. One such technique uses a release history buffer).

As regards to Claim 5, *A method according to claim 1 comprising, at the first network interface, determining a size of the data and, based upon the size of the data, selecting among two or more protocols for transmitting the data* (Conway as stated in col. 8, lines 51-59, discloses, In the illustrated embodiment the ordered procedure for granting fair access to the lock variable was in round robin order. In other embodiments other ordered procedures for fair access are possible. Examples include reverse round robin order, an arbitrary order among all the nodes, historically balanced order, and random order. Note also that while the inter-node communication links were formed by coherent HyperTransport, other communication protocols may be used as well).

Conway doe not disclose selecting protocol based on the size of data.

In the same field of multiprocessor data processing system Franke discloses A method, system, and associated program code and data structures are provided for a message processing system in which messages are transmitted from source nodes to destination nodes. The multiprocessor data processing system 10 depicted in FIG. 1, require reliable message communication paths between respective ones of the processors 12.sub.1 . . . 12.sub.j. The exemplary system 10 of FIG. 1 employs an exemplary communication medium or switch network 20 commonly coupled to the processors 12. The processors may require respective communication adapters

14.sub.1 . . . 14.sub.j to control communications between each processor 12 and the medium 20 via respective connections 16.sub.1 . . . 16.sub.j. Further as illustrated in FIG. 3, the software and/or hardware of destination node 18.sub.j includes early arrival processing resources 52 and pre-allocated, early arrival buffering 50 of sufficient size to accommodate length "N" data portions of a predetermined number "Q" of the messages from each of "j-1" potential message source nodes which may require receipt at the destination node.

It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to modify Conway teachings of non-uniform memory access (NUMA) architecture, in which each processor operates on a shared address space but memory is distributed among the processor nodes and memory access time depends on the location of the data in relation to the processor that needs it and to incorporate the teachings of Franke in a multiprocessor message processing system including protocols and buffering, for facilitating the transmission of messages from a source node to a destination node.

The modifications would have been obvious because one of ordinary skill in the art would have been motivated for a method, system, and associated program code and data structures, protocols and buffering for facilitating the efficient transmission of messages from a source node to a destination node in a message processing system as suggested by Frank and Conway open to vast number of variations which exist. Examples include reverse round robin order, an arbitrary order among all the nodes, historically balanced order, and random order. Note also that while the inter-node

communication links were formed by coherent HyperTransport, other communication protocols may be used as well.

Hence together Conway and Franke disclose all limitation of Claim 5 dependent on Claim 1.

As regards to Claim 6, *A method according to claim 5 wherein the two or more protocols comprise an eager protocol and a rendezvous protocol* (Conway and Franke disclose all limitation of Claim 5. Further Franke as stated in col. 4, lines 54-60 discloses, FIG. 5 is a protocol diagram of a second, eager rendezvous transmission mode in which message transmission is initiated using a packet having both control information and a data portion of the message, with any remaining data portions of the message being transmitted following an acknowledgement from the destination node).

As regards to Claim 7, *A method according to claim 6 comprising, upon selecting the rendezvous protocol, automatically generating a Ready To Send message at the first network interface of the first compute node* (Conway and Franke disclose all limitation of Claim 6. Further Franke as stated in col. 6, lines 58-67, col. 7, lines 1-17 discloses, With reference to the protocol diagram of FIG. 4, shown therein is a first rendezvous transmission mode 100 for transmitting message M(1), 102 between source node 18.sub.1, and destination node 18.sub.j via medium 20. Message M(1) 102 is assumed to have multiple data portions, i.e., P1, P2 and P3. Upon a determination in source node 18.sub.1 to send message M(1), step 104, "SEND ID, L" occurs in which

initial packet 106 is transmitted from the source node to the destination node. In mode 100, only this control information is initially sent from the source to the destination. (In mode 100, and the modes discussed below in connection with FIGS. 5 and 6, the destination always guarantees sufficient space for control information, since its size is negligible.) In response to receiving this information at the destination, inquiry 108, "READY TO RECEIVE M(1)?" occurs, during which a determination is made whether, for example, adequate receive buffering is available at the destination to receive message M(1) having length L.sub.M(1)).

As regards to Claim 8, *A method according to claim 1 wherein the data comprises a raw ethertype datagram and transmitting the data comprises encapsulating the raw ethertype datagram within one or more link layer packet headers.*

Conway does not disclose data comprising raw ethertype datagram.

Conway does disclose as stated in col. 5, lines 6-16, HT link controllers 308, 310, and 312 are coupled to devices external to microprocessor 300 over corresponding input and output channels. Each of HT link controllers 308, 310, and 312 complies with the physical interface specified in the HyperTransport I/O Link Specification. In node 102 HT link controllers 308 and 310 function as coherent links that communicate with nodes 101 and 104 of FIG. 1, respectively, using a coherent form of HyperTransport. HT link controller 312, functions as a host bridge that communicates with I/O device 340 using the non-coherent form of HyperTransport.

In an analogues art of communications between multi-processor clusters of multi-cluster computer systems Kota discloses multiple cluster, multiple processor system where each processing cluster 121, 123, 125, and 127 is coupled to a switch 131 through point-to-point links 141a-d. A switch 131 can include a general purpose processor with a coherence protocol interface and where coherence protocol interface is implemented by Conway also.

Further as stated in par. [0093-0094], Kota discloses, The interconnection controller has a coherent protocol interface 307 having an intra-cluster interface that allows the interconnection controller to communicate with other processors in the cluster via intra-cluster links such as 232a-232d of FIG. 2. Coherent protocol interface 307 also includes an inter-cluster interface that allows interconnection controller 230 to communicate with external processor clusters via, for example, links 111a-111f of FIGS. 1A and 1B. The interconnection controller may also include other interfaces such as a non-coherent protocol interface 311 for communicating with I/O devices (e.g., as represented in FIG. 2 by links 208c and 208d). An interconnection controller will preferably have multiple serializers/deserializers 313 and transceivers 315. According to some implementations, before a packet from a local processor is forwarded to an interconnection controller in another cluster, a remote transceiver 315 processes the packet. The processing may include adding a packet header and adding cyclic redundancy code check information. A remote serializer/deserializer 313 serializes packets to be sent on an inter-cluster link and deserializes packets received from an

inter-cluster link. A remote serializer/deserializer 313 preferably performs 8b/10b encoding and 10b/8b decoding (or comparable encoding and decoding).

It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to modify Conway teachings of HT link controllers function as coherent links that communicate with nodes and functions as a host bridge that communicates with I/O device using the non-coherent form of HyperTransport, and to incorporate the teachings of Kota's interconnection controller coherent protocol interface and non-coherent protocol interface communicate with external processor clusters, which improves upon the teachings of Conway.

Hence together Conway and Kota disclose all limitation of Claim 8 dependent on Claim 1.

As regards to Claim 9, *A method according to claim 8 wherein the link layer packet headers comprise InfiniBand TM link layer packet headers* (Conway and Kota disclose all limitation of Claim 8. Further Kota as stated in par. [0231], discloses, FIG. 16 depicts a process for detecting and eliminating skew according to some implementations. In step 1605, an inter-cluster initialization sequence is performed. The inter-cluster initialization sequence may include, for example, the use of one or more training sequences having known structures and lengths. These training sequences may be novel or may be analogous to training sequences employed in other contexts, such as, for example, TS1 and TS2 of the InfiniBand.TM. protocol. The InfiniBand Architecture Release 1.1, dated Nov. 6, 2002, particularly Section 5.6.1, "Link De-Skew

and Training Sequence," is hereby incorporated by reference. The training sequences may be repeated on each data lane of the inter-cluster link until all data lanes are synchronized, e.g., until a phase-locked loop ("PLL") is established for the transmitting and receiving interconnection controllers on each data lane of the inter-cluster link).

As regards to Claim 10, *A method according to claim 1 wherein the data comprises a raw internet protocol datagram and transmitting the data comprises encapsulating the internet protocol datagram within one or more link layer packet headers* (Conway and Kota disclose all limitation of Claim 8 dependent on claim 1. Conway as stated in col. 5, lines 17-22 discloses, I/O device 340 is an input/output device that, for example, implements the local area network communication protocol standardized by the Institute of Electrical and Electronics Engineers (IEEE) under the auspices of the IEEE 802.3 committee, commonly referred to as "Ethernet". However other types of I/O functions are possible as well. Further Kota as stated in par. [0089], [0163-0165], discloses, I/O adapter 220 may be an Ethernet card adapted to provide communications with a network such as, for example, a local area network (LAN) or the Internet. In step 1120, a packet (in this example, an HT packet) is received by an interconnection controller in a home cluster via an intra-cluster link. The packet may be received, for example, from a processor within the home cluster according to a coherent protocol. Alternatively, the packet may be received from an I/O device in the home cluster via a noncoherent protocol. In step 1135, a header is added to the packet. Preferably, the header is used for link layer encapsulation of the packet, transforming

the packet to a high-speed link ("HSL") packet. In step 1140, a CRC check is performed based only upon the HSL packet and its header).

As regards to Claims 11-12, 15, 16, 17-18 and 19 *lists all the same elements of* ***Claims 1-2, 8, 4, 3 and 5****, but in an compute node for use in a multi-compute-node computer system form rather than method form respectively. Therefore, the supporting rationale of the rejection to Claims 1-2, 8, 4, 3 and 5, applies equally as well to Claims 11-12, 15, 16, 17-18 and 19.*

As regards to Claim 13, *A compute node according to claim 11 comprising a memory, and a facility configured to allocate eager protocol buffers in the memory and to automatically signal to one or more other compute nodes that the eager protocol buffers have been allocated* (Conway and Franke disclose all limitation of Claim 11. Further Franke as stated in col. 5, lines 50-67, col. 6, lines 1-10, as illustrated in FIG. 3, a predetermined number "Q" of early arrival buffer slots, each of length "N," are provided at the destination node for each of "j-1" potential message source nodes so that at least a portion of the data of at least some of the messages 30 can be accommodated at the destination node along with the initial control information. Relying on the presence of the receive buffering, or in the alternative the early arrival buffering 50 at the destination node, the source node, in an "eager" rendezvous mode, can reliably send at least some data portions of at least some of the messages 30 with the respective initial transmissions of control information. (If the overall message size is

smaller than "N," the entire message is transmitted.) When the destination is ready, e.g., adequate receive buffering is posted for the entirety of the message, the destination node copies 46 the data from the early arrival buffering to the posted receive buffering, sends a respective acknowledgement to the source node, and the source node can at that time send remaining portions of the data of the respective, now acknowledged messages. In the case of small messages "MS(i)" of size less than N, i.e., those that fit into a single data portion (frame), the acknowledgement shall not be understood as direction to continue sending data (since all data was sent in MS(i)), nor as an indication that the source sending MS(i) has to wait, but rather that an early arrival buffer has become free (see discussions below regarding how the acknowledgements are used herein to indicate the freeing of early arrival buffer space).

As regards to Claim 14, *A compute node according to claim 13 comprising a facility configured to automatically associate memory protection keys with the eager protocol buffers and a facility configured to verify memory protection keys in incoming eager protocol messages before writing the incoming eager protocol messages to the eager protocol buffers* (Conway and Franke disclose all limitation of Claim 13. Further Franke as stated in col. 6, lines 11-45, For these small messages, since acknowledgements are costly to send, the acknowledgements for multiple such small messages can be grouped together, and sent as a single packet, possibly combined with other control information. If the destination withholds these acknowledgements for too long, however, the source may revert to standard rendezvous, as discussed below

in connection with FIG. 6. One possibility is to impose a high-water mark at the destination to ensure that the source does not mistakenly assume that early arrival buffering is filled, when in fact it is free, but the destination is intentionally withholding acknowledgements. When the high-water mark is reached, the grouped acknowledgements would then be sent, thus indicating to the source the correct status of the early arrival buffering. Also, explicit acknowledgements may not be necessary for each message, since the number of such messages can be returned by the destination (if a certain high-water mark has been reached) and this number suffices as the acknowledgement for each message. The length "N" of the first data portion of the message transmitted is a predetermined number which should correspond to the size "N" of the early arrival buffer slot pre-allocated at the destination node for each message of a number "Q" of messages. If "N" is large enough so that the destination node receives the control information in the initial transmission and returns the rendezvous acknowledgement before all "N" bytes have been sent by the source node, then the source node does not experience any interruption in the data transmission and the destination, likewise, does not see any interruption in the data received. This eager rendezvous transmission mode does not require a large amount of buffering at the destination for these initial transmissions since each such early arrival transmission only brings with it, at most, "N" bytes of the data portion of the message).

As regards to Claim 20, *lists all the same elements of* ***Claim 1****, but in a computer system comprising a plurality of compute nodes according to claim 11,*

instead of method from. Therefore, the supporting rationale of the rejection to Claim 1, applies equally as well to Claim 20.

Remarks

5. The following pertaining arts are discovered and not used in this office action. Office reserves the right to use these arts in later actions.

a. Gonzalez; Ricardo E. et al. (US 7613900 B2) Systems and methods for selecting input/output configuration in an integrated circuit

b. Hooper, Donald F. et al. (US 20040233934 A1) Controlling access to sections of instructions

c. Rowlands; Joseph B. et al. (US 6944719 B2) Scalable cache coherent distributed shared memory processing system

d. Conway; Patrick (US 7062610 B2) Method and apparatus for reducing overhead in a data processing system with a cache

Conclusion

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Muktesh G. Gupta whose telephone number is 571-270-5011. The examiner can normally be reached on Monday-Friday, 8:00 a.m. -5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William C. Vaughn can be reached on 571-272-3922. The fax phone

number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/William C. Vaughn, Jr./

Supervisory Patent Examiner, Art Unit 2444

MG